Metastorm BPM®
Release 7.6

DMS Provider Framework Developer Guide

May 2008

Metastorm Inc.

email: inquiries@metastorm.com

http://www.metastorm.com

Copyrights and Trademarks

© 1996-2008 Metastorm Inc. All Rights Reserved.

Copyright Notice

- Metastorm®, Metastorm BPM®, e-Work®, Process Pod®, Enterprise Process Advantage®, ProVision®, The Best Process Wins®, Proforma®, Metastorm Knowledge Exchange™, Metastorm DNA™, Metastorm Discovery™, STAR™, Insight™, Envision™, and Metastorm Enterprise™ are either registered trademarks or trademarks of Metastorm in the United States and/or other countries.
- Microsoft®, Outlook®, SQL Server™, Windows®, Vista™, Active Directory®, Visual Basic®, JScript®, SharePoint® and BizTalk® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
- Adobe® is a registered trademark of Adobe Systems, Inc.
- Netscape® is a registered trademark of Netscape Communications Corporation.
- Other trademarks are the property of their respective owners.

Disclaimer

Every effort has been made to ensure the accuracy of the features and techniques presented in this publication. However, Metastorm accepts no responsibility, and offers no warranty whether expressed or implied, for the accuracy of this publication.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the express written permission of Metastorm, Inc.

The information in this document is subject to change without notice.

Metastorm Inc.

email: inquiries@metastorm.com

http://www.metastorm.com

Metastorm BPM Release 7.6

DMS Provider Framework Developer Guide

Contents

1 Introduction 5
1.1 Acronyms 5
1.2 Getting Further Information 6
2 What is the Metastorm DMS Provider Framework? 7
2.1 Architecture 8
2.2 Metastorm SSO 8
2.3 Metastorm Web Client 9
2.4 Metastorm Web Extensions 9
2.5 DMS Provider 9
3 Getting Started 10
3.1 Files and Dependencies 10
3.1.1 Metastorm.Dms.dll 10
3.1.2 Metastorm.Dms.SharePoint.dll 11
3.1.3 DMS Common Library.xel 11
3.1.4 SharePoint DMS Library.xel 11
4 Writing a DMS Provider 12
4.1 Document Identifier 12
4.2 Metastorm DMS Schema 12
5 Deployment 14
5.1 DMS Provider Assembly 14
5.1.1 Engine Machine 14
5.1.2 Web Extensions Machine 15
5.2 DMS Provider Library 16
Appendix A - The DMS Provider Interface 17
Declaration Syntax 17
Methods 17
Properties 18
Events 18
Appendix B - The DMS Provider Library 19
Upload Document 19
Browse DMS 20
Check In Document 20
Check Out Document 20
Delete Document 21
Download Document 21
Get Document Metadata 22
Set Document Metadata 22
Undo Document Check Out 23

Verify Document 23
Verify Folder 24
Appendix C – Metastorm DMS Schema 25

Metastorm BPM Release 7.6

DMS Provider Framework Developer Guide

1 INTRODUCTION

This guide is intended to describe:

- What the Metastorm DMS Provider Framework is.
- How to create a DMS provider to use in Metastorm BPM.
- How to deploy a DMS provider in Metastorm BPM.

1.1 Acronyms

The following table lists the acronyms used in this guide:

Acronym	*Meaning*
DMS	Document Management Support
COM+	Component Object Model
HTTP	HyperText Transfer Protocol
MOSS	Microsoft Office SharePoint Server
SSO	Windows Single Sign-On

Table 1: Acronyms

1.2 Getting Further Information

The following table lists where to find detailed information on these topics:

For Information on	*See*
Metastorm DMS Provider Framework Functions	DMSProviderFramework.chm
DMS Multi clip and Single Clip Client Guide	Metastorm BPM Concepts
Designing DMS Multi clip and Single Clip	Designer User Manual
DMS Installation Prerequisites	Installation Prerequisites Guide
DMS Configuration	Adminsitration Guide

Table 2: Getting further information

 © Metastorm Inc., 2008

2 WHAT IS THE METASTORM DMS PROVIDER FRAMEWORK?

The Metastorm DMS Provider Framework is a concept that provides a generic way of accessing and managing documents in Document Management Support (DMS) systems from within Metastorm BPM.

Support for individual DMS is provided in the form of the following components:

- DMS Provider Assembly – This is a .NET assembly that implements the `Metastorm.Dms.IMetastormDmsProvider` interface.
- DMS Provider Designer Library – This is a Metastorm Designer library, activated using the Metastorm Process Orchestrator for .NET, and exposes the DMS functionality in the DMS Provider Assembly in the form of Metastorm Designer Integration Wizard functions.

2.1 Architecture

The following diagram illustrates how a DMS Provider assembly interacts with the other components in Metastorm BPM:

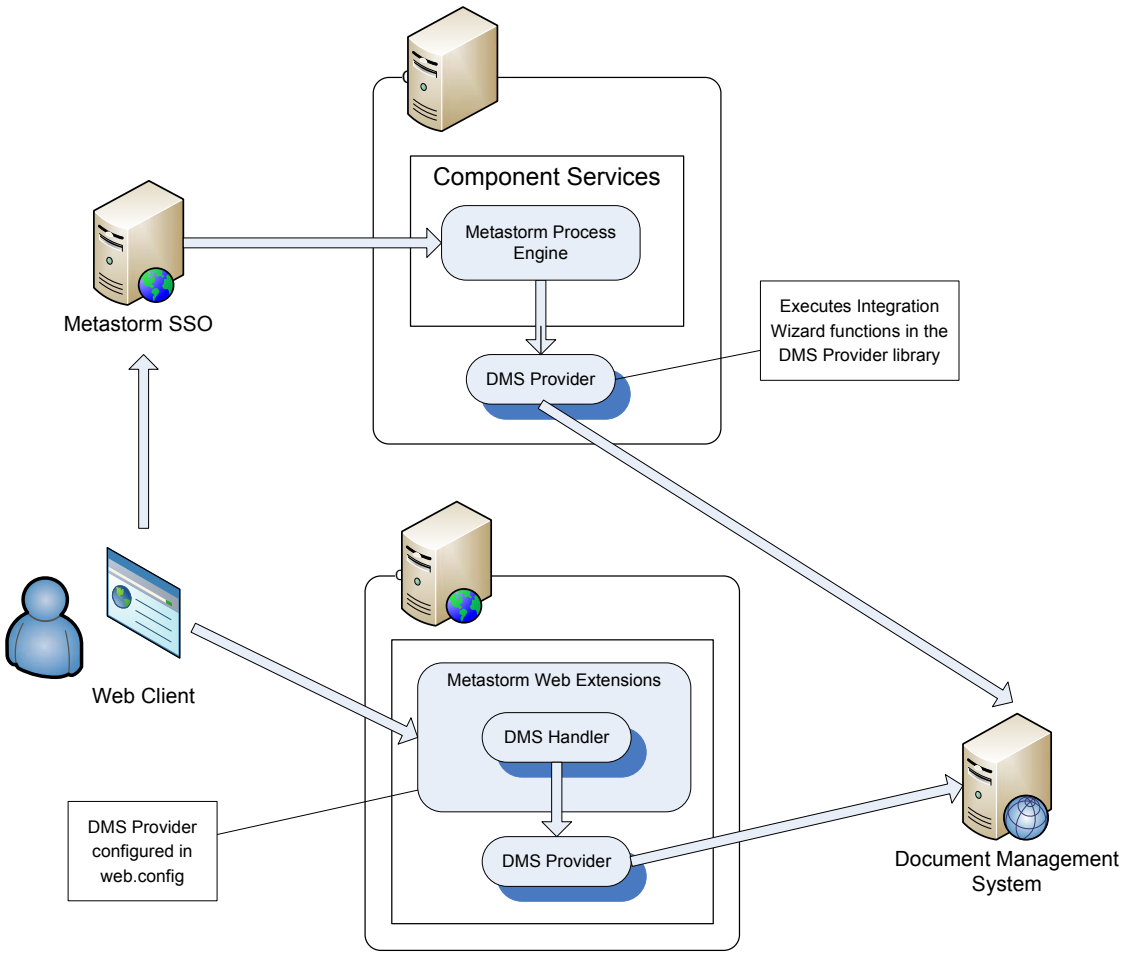

Figure 1: Document Management Support in Metastorm BPM

2.2 Metastorm SSO

In order to flow users credentials all the way to the DMS Provider, it is envisaged that Metastorm BPM will be configured in a Windows Single Sign-On (SSO) environment.

In the Metastorm Web Client, all calls to the DMS Provider will be made using the current logged on user's credentials.

 © Metastorm Inc., 2008

For DMS Provider Library functions executing in the Metastorm Process Engine, these calls will be made using the **Metastorm Process Engine** COM+ application identity.

The DMS Provider must deal with any credential mapping if the DMS does not support Windows SSO.

2.3 Metastorm Web Client

DMS support in the Metastorm Web Client is provided through the new DMS clip field and DMS multi-clip field in the Metastorm BPM forms.

2.4 Metastorm Web Extensions

A new HTTP Handler, `eDms.ashx`, has been added to the Metastorm Web Extensions to enable the Metastorm web client to communicate with the DMS using the configured DMS Provider.

2.5 DMS Provider

This .NET assembly implements the `Metastorm.Dms.IMetastormDmsProvider` interface to enable the Metastorm Web Client and Metastorm Process Engine to talk to DMS.

3 GETTING STARTED

3.1 Files and Dependencies

The following files are delivered as part of the **Metastorm Document Management Support System** installation feature:

```
/Engine/dotnetbin/
      Metastorm.Dms.dll
/Web/bin/
      Metastorm.Dms.dll
      Metastorm.Dms.SharePoint.dll
/Administrative Tools/Administrative Procedures/
      DMS Common Library.xel
      SharePoint DMS Library.xel
```

Typically these files can be found in the default location: `C:\Program Files\Metastorm BPM`.

3.1.1 Metastorm.Dms.dll

This is the DMS Provider Framework base assembly. The mandatory `Metastorm.Dms.IMetastormDmsProvider` interface that every DMS provider must implements is defined in this assembly.

 © Metastorm Inc., 2008

3.1.2 Metastorm.Dms.SharePoint.dll

This is the Microsoft Office SharePoint Server (MOSS) 2007 DMS provider that ships with Metastorm BPM Release 7.6.

3.1.3 DMS Common Library.xel

This is a Metastorm Designer library containing the following Integration Wizard functions to help Process Designers to create procedures to invoke DMS commands in the Process Engine via server script:

- **Get Document Id From Clip**: Retrieves the document id in a DMS clip.
- **Get Selected Document Id**: Returns the document id of the currently selected document in a DMS clip field or DMS multi-clip field.
- **Get Folder Document Ids**: Returns a delimited list of document ids attached to the current folder

3.1.4 SharePoint DMS Library.xel

Each specialized DMS provider must be accompanied with its own Metastorm Designer library to enable process designers to invoke DMS commands in the Metastorm Process Engine via server scripts. The SharePoint DMS designer library file, `SharePoint DMS Library.xel`, has the following Integration Wizard functions, some are mandatory and some are specific to SharePoint:

Mandatory Functions

See Appendix B for details.

SharePoint Specific Functions

- **Get Document Author**: Gets the author (creator) of a specified document.
- **Get Document Checked Out By**: Gets the user who currently has a specified document checked out.
- **Get Document Creation Date**: Gets the date when a specified document is created.
- **Get Document Editor**: Gets the editor (modified by) of a specified document.
- **Get Document Last Modified**: Gets the date when a specified document is last modified.
- **Get Document Title**: Gets the title of a specified document.

4 WRITING A DMS PROVIDER

Writing a DMS Provider for Metastorm BPM is fairly straightforward. The two components to implement are the DMS Provider Assembly and the DMS Provider Designer Library. Let's look briefly at each of these components.

- **DMS Provider Assembly**. This is a .NET assembly that implements the mandatory `Metastorm.Dms.IMetastormDmsProvider` interface, see Appendix A for details.

- **DMS Provider Designer Library**. This is a Metastorm Designer library, activated using the Metastorm Process Orchestrator for .NET, that exposes the DMS functionality in the DMS Provider Assembly in the form of Metastorm Designer Integration Wizard functions, see Appendix B for details.

4.1 Document Identifier

Most of the methods defined in the `IMetastormDmsProvider` interface require a document identifier as an input parameter. A document identifier is simply a string value to enable the DMS provider to identify the document in the DMS document repository. For the SharePoint DMS Provider, the document identifier is a URL, but other DMS Provider implementations are free to use whatever they want to identify documents in the DMS document repository.

4.2 Metastorm DMS Schema

With the exception of **BrowseDms**, most of the methods in the `IMetastormDmsProvider` interface are fairly straightforward. The **BrowseDms** method takes a string value, representing a location in the DMS repository, as input and returns an XML fragment to represent the specified

 © Metastorm Inc., 2008

location as output. The XML fragment the **BrowseDms** method must conform to the Metastorm XML schema, see Appendix C for details.

As the DMS support in Metastorm BPM is designed to be generic, a simple folder and file hierarchy has been adopted to represent DMS document repositories. The XML fragment below is a sample output from the SharePoint DMS Provider:

```
<?xml version="1.0" encoding="utf-16"?>
<DMSStore Location="http://localhost/Lists/Links">
<DmsItem
  Name="AllItems.aspx"
  Location="http://localhost/Lists/Links/AllItems.aspx" />
  <DmsItem
    Name="DispForm.aspx"
    Location="http://localhost/Lists/Links/DispForm.aspx" />
  <DmsItem
    Name="EditForm.aspx"
    Location="http://localhost/Lists/Links/EditForm.aspx" />
  <DmsItem
    Name="NewForm.aspx"
    Location="http://localhost/Lists/Links/NewForm.aspx" />
  <DmsContainer
    Name="Link"
    Location="http://localhost/Lists/Links/Link" />
</DMSStore>
```

In the above sample, the `<DmsItem>` element represents a document and the `<DmsContainer>` element represents a part of the DMS repository that can hold documents within its hierarchy. For SharePoint, a `<DmsContainer>` element could represent, a site, list, document library or a library folder.

5 DEPLOYMENT

This section provides details on deploying custom DMS providers that conform to the Metastorm DMS Provider Framework in Metastorm BPM.

5.1 DMS Provider Assembly

As illustrated in Figure 1, both the Metastorm Process Engine and the Metastorm Web Extensions depend on the DMS Provider to deal with DMS related requests. Therefore, additional DMS providers must be deployed in the following locations:

- Engine Machine
- Web Extensions Machine

5.1.1 Engine Machine

The Metastorm Process Engine uses its existing Process Orchestrator for .NET to process DMS related requests. As such, all custom DMS Providers and dependent assemblies must be deployed in the Metastorm Process Engine's JScript.NET library path. The Metastorm Process Engine's JScript.NET library path is defined in the following registry setting:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Metastorm\e-work\Engine\JScriptLib
```

The default value for the above registry key is as follows:

```
C:\Program Files\Metastorm BPM\Engine\dotnetbin;C:\Program
Files\Metastorm BPM\Engine\dotnetlib
```

 © Metastorm Inc., 2008

The out of the box SharePoint DMS Provider is deployed in the `dotnetbin` directory. It is recommended that custom DMS Provider assemblies are deployed in the same location.

*There is no need to install Metastorm Process Orchestrator for .NET to use the new DMS support feature. The Metastorm BPM core installation has been updated to deploy all the dependent files if the **Metastorm Document Management Support** option is selected.*

5.1.2 Web Extensions Machine

The Metastorm Web Extensions use a configured DMS provider to execute most of the document commands in the new DMS clip and DMS multi-clip fields. The configured DMS provider is also used to render the DMS document repository browser window. If the **Metastorm Document Management Support** option is selected during installation then a SharePoint DMS provider will be configured and deployed, by default, in the following location:

```
C:\Program Files\Metastorm BPM\Web\bin
```

Any custom DMS provider and dependent assemblies must also be deployed in the above location.

The Metastorm Web Extensions are designed to support any DMS provider that conforms to the Metastorm DMS Provider framework. This is achieved through a web configuration setting, illustrated in the following Metastorm Web Extensions' `web.config` file snippet:

```
<configuration>
  <configSections>
    ...
    <section
      name="dms"
      type="Metastorm.Dms.MetastormDmsSection, Metastorm.Dms" />
  </configSections>
  ...
  <!-- SharePoint DMS Provider -->
  <dms provider="Metastorm.Dms.SharePoint,
                 Version=7.0.0.0,
                 Culture=neutral,
                 PublicKeyToken=0fa3cc64eebf4c8b"
       type="Metastorm.Dms.SharePoint.SPDmsProvider" />
</configuration>
```

If the **Metastorm Document Management Support** option is selected during installation then the above highlighted settings will be added to the Metastorm Web Extensions' `web.config` file. The Metastorm Web Extensions is designed to run only one DMS provider at a time. To support a custom DMS provider, the `<dms>` element in the `web.config` file must be reconfigured as follows:

1. Specify the custom DMS provider's full assembly name in the **provider** attribute value.

2. Specify the DMS provider class name in the **type** attribute. The DMS provider class is the .NET class the implements the `Metastorm.Dms.IMetastormDmsProvider` interface.

5.2 DMS Provider Library

The custom DMS provider designer library should be copied to the following directory on the Metastorm Designer machine, along with the other Metastorm administrative procedures:

```
C:\Program Files\Metastorm BPM\Administrative
Tools\Administrative Procedures
```

The above location assumes Metastorm BPM is installed in the default C: drive.

 © Metastorm Inc., 2008

APPENDIX A - THE DMS PROVIDER INTERFACE

Each specialized DMS Provider must implement the mandatory interface, `Metastorm.Dms.IMetastormDmsProvider`.

Declaration Syntax

```
public interface IMetastormDmsProvider
```

Methods

Member	*Description*
AddFile(Byte[], String, String)	Adds a new file to the document store.
AddFile(String, String, String)	Adds a new file to the DMS form the local file system.
BroweseDms(String)	Allows the user to browse a DMS location. The XML must comfort with the DmsProviderStoreSchema.
CheckIn(String, String)	Check In an already checked out file.
CheckOut(String)	Check Out a file for editing.
DeleteFile(String)	Delete a file from the store.
GetDocument(String, out String)	Retrieves a file from SharePoint.
GetDocument(String, String)	Retrieves a file from the DMS and saves it at the specified path.
GetDocumentMetaData(String, String)	Retreives the a single metadata value of a delimited list of metadata values for a file.
SetDocumentMetaData(String, String, String)	Updates document metadata with the specified value.
UndoCheckOut(String)	Removes the checked out flag from a document.
UpdateFile(Byte[], String)	Updates a file in the document store.

VerifyFile(String)	Verifies that the specified file exists.
VerifyFile(String, String)	Verifies that the specified file exists. Overloaded method to allow the document location and file name to specify separately.

Properties

Member	*Description*
Description	Gets the description of the provider.
IsDirectlyBrowsable	Gets whether the Dms can be directly accessed outside the Provider.
Name	Gets the name of the provider.
SupportedDms	Gets the name of the supported DMS.

Events

Member	*Description*
Error	Raised when an operation error occurs.
FileAdded	Raised when a file is added to the document store.
FileCheckedIn	Raised when a file is checked in the document store.
FileCheckedOut	Raised when a file is checked out from the document store.
FileDeleted	Raised when a file is deleted from the document store.
FileUndoCheckedOut	Raised when a file's check out state is discarded.

*Please refer to the **Metastorm DMS Provider Framework** reference document for a full description of the IMetastormDMSProvider interface.*

 © Metastorm Inc., 2008

APPENDIX B - THE DMS PROVIDER LIBRARY

Each DMS Provider must be accompanied by its own Metastorm Designer library to enable Process Designers to use the Integration Wizard to create DMS commands to execute in the Metastorm Process Engine. The Integration Wizard functions DMS Provider library must implement, at a bare minimum, the following functions:

Upload Document

This function is used to upload a document to the Document Management System (DMS).

Arguments

Display Name: Server File Name.
Type: Text.
Hint: The file name as it will be saved on the server.
Required: True.

Display Name: Local File.
Type: Text.
Hint: The full path to the file to upload.
Required: True.

Display Name: Server Location.
Type: Text.
Hint: The location in the DMS where the document is to be uploaded.
Required: True.

Return Value

Type: Text.

A document identifier allocated by the DMS.

Browse DMS

This function returns the content of a specified DMS location in XML format.

Arguments

Display Name: DMS Location.
Type: Text.
Hint: A location within the DMS repository.
Required: True.

Return Value

Type: Text
An XML fragment to represent the specified DMS location.

Check In Document

This function enables a document to be checked back into the DMS.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier .
Required: True.

Display Name: Comments.
Type: Text.
Hint: Comments to include with the document being checked in.
Required: True.

Return Value

Type: Integer.
A value of 0 if sucessful; otherwise 1.

Check Out Document

This function enables a document to be checked out from the DMS.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier.
Required: True.

Return Value

Type: Text.
A value to indicate success or failure.

Delete Document

This function deletes a document from the DMS.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier .
Required: True.

Return Value

Type: Integer.
A value of 0 if sucessful; otherwise 1.

Download Document

This function downloads a specified document to the local file system.

Arguments

Display Name: File Name.
Type: Text
Hint: A string containing the document identifier .
Required: True.

Display Name: Local File Name.
Type: Text.
Hint: The full path where the document is to be saved on the local file system .
Required: True.

Return Value

Type: Integer.
A value of 0 if sucessful; otherwise 1.

Get Document Metadata

This function retrieves the value(s) of the specified document metadata.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier .
Required: True.

Display Name: Metadata Names.
Type: Text.
Hint: A comma-separated list of document metadata names.
Required: True.

Return Value

Type: Text.
A comma-separated list of document metadata values.

Set Document Metadata

This function assigns a value to a specified document metadata.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier.
Required: True.

Display Name: Metadata Name.
Type: Text.
Hint: The name of the document metadata to modify.
Required: True.

Display Name: Custom Variable.
Type: Option.
Hint: Selects the custom variable whose value is to be assigned to the document metadata.
Required: True.

Return Value

Type: Integer.

A value of 0 if sucessful; otherwise 1.

Undo Document Check Out

This function undoes a document check out.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier.
Required: True.

Display Name: Metadata Name.
Type: Text.
Hint: The name of the document metadata to modify.
Required: True.

Display Name: Custom Variable.
Type: Option.
Hint: Selects the custom variable whose value is to be assigned to the document metadata. .
Required: True.

Return Value

Type: Integer.
A value of 0 if sucessful; otherwise 1.

Verify Document

This function determines if a specified document exists in the DMS repository.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the document identifier.
Required: True.

Return Value

Type: Check.
A value to indicate if the specified document exists.

Verify Folder

This function determines if a specified location exist in the DMS repository.

Arguments

Display Name: File Name.
Type: Text.
Hint: A string containing the DMS repository location.
Required: True.

Return Value

Type: Check.
A value to indicate if the specified location exists in the DMS repository.

 © Metastorm Inc., 2008

APPENDIX C – METASTORM DMS SCHEMA

The Metastorm DMS provider framework represents DMS repositories internally in a hierarchical structure that conforms to the following XML schema:

```
<?xml version="1.0" encoding="utf-8"?>
  <xs:schema id="DmsProviderSchema"
targetNamespace="http://schemas.metastorm.com/DmsProviderBrowse.
xsd"   elementFormDefault="qualified"
xmlns="http://schemas.metastorm.com/DmsProviderBrowse.xsd"
xmlns:mstns="http://schemas.metastorm.com/DmsProviderBrowse.xsd"
xmlns:xs="http://www.w3.org/2001/XMLSchema" version="7.6">
    <xs:element name="DMSStore">
      <xs:complexType>
        <xs:sequence>
          <xs:element ref="DmsContainer" maxOccurs="unbounded"
minOccurs="0" />
          <xs:element ref="DmsItem" maxOccurs="unbounded"
minOccurs="0" />
        </xs:sequence>
        <xs:attribute name="Location" type="xs:string" />
      </xs:complexType>
    </xs:element>
    <xs:element name="DmsItem">
    <xs:complexType>
      <xs:sequence />
        <xs:attribute name="Name" type="xs:string" />
        <xs:attribute name="Location" type="xs:string" />
    </xs:complexType>
```

```
  </xs:element>
  <xs:element name="DmsContainer">
    <xs:complexType>
      <xs:sequence>
      </xs:sequence>
      <xs:attribute name="Location" type="xs:string" />
      <xs:attribute name="Name" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

 © Metastorm Inc., 2008